# Mining transport data for Carbon emissions

Exploring the Role of Transport Data in Achieving SDG 13 Climate Action

# Mining transport data for carbon emissions

**MINI PROJECT REPORT**


**SUBMITTED BY**

**NAVYA SREE.D(Reg.No 814723104103)**

**NAZEER HUSSAIN.S(Reg.No 814723104104)**


**Course code:CCS334**

**Course name:Big Data Analysis**

**Department:**CSE-B

***BACHELOR OF ENGINEERING***

*in*

*COMPUTER SCIENCE AND ENGINEERING*

***SRM TRP ENGINEERING COLLEGE,***

***TIRUCHIRAPALLI***

***NOV/DEC-2025***

| S. No. | Content |
|---|---|
| 1 | **Abstract** |
| 2 | **Introduction** |
| 2.1 | Overview of the Topic — Mining Transport Data for Carbon Emissions under SDG 13 (Climate Action) |
| 2.2 | Problem Definition — Rising $CO_2$ emissions from transport systems and lack of predictive data analytics |
| 2.3 | Objectives of the Project — Analyze and predict carbon emissions using simulated transport data |
| 2.4 | Scope and Significance — Supports SDG 13 by identifying emission trends and promoting sustainable mobility |
| 3 | **Literature Survey** |
| 3.1 | Related Existing Systems or Research Work — Prior studies on vehicular emission analysis and data mining |
| 3.2 | Limitations of Existing Systems — Limited interpretability, lack of real-time analysis, and poor scalability |
| 3.3 | Proposed System Advantages — Lightweight analytics, visualization, and simulation without external datasets |
| 4 | **System Analysis** |
| 4.1 | Problem Identification — Estimating $CO_2$ emissions based on vehicle characteristics |
| | |

| | |
|---|---|
| 4.2 | Feasibility Study |
| 4.2.1 | Technical Feasibility — Implemented in Python using libraries like Pandas, Matplotlib, and Seaborn |
| 4.2.2 | Operational Feasibility — Easy to execute on local systems or Jupyter Notebook |
| 4.2.3 | Economic Feasibility — No external cost; simulated dataset used |
| 4.3 | Requirements Specification |
| 4.3.1 | Functional Requirements — Dataset generation, analysis, visualization, emission prediction |
| 4.3.2 | Non-Functional Requirements — Accuracy, efficiency, portability, and maintainability |
| 5 | **System Design** |
| 5.1 | System Architecture Diagram — Data generation, preprocessing, analysis, visualization flow |
| 5.2 | Data Flow Diagram (DFD) — Shows data movement from input to visualization and output prediction |
| 5.3 | UML Diagrams (Use-Case, Class, Sequence, Activity) — Depict relationships between modules and data |
| 5.4 | Database Design (ER Diagram) — Attributes for Vehicle ID, Engine Size, Fuel Type, $CO_2$ output |
| 5.5 | Module Descriptions — Dataset Simulation, Data Analysis, Visualization, Model Evaluation |

| | |
|---|---|
| 6 | **Implementation** |
| 6.1 | Tools and Technologies Used — Python, NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn |
| 6.2 | Modules Implemented — Synthetic data generation, $CO_2$ visualization, regression modeling |
| 6.3 | Code Snippets — Essential Python code sections for execution and visualization |
| 7 | **Testing and Validation** |
| 7.1 | Testing Strategy — Manual verification and regression model validation |
| 7.2 | Test Cases and Results — Comparison of predicted and actual emission values |
| 7.3 | Validation Outcomes — Model accuracy metrics and visual confirmation |
| 8 | **Results and Discussion** |
| 8.1 | Output Screenshots — Graphs of emission distribution, fuel-type comparison, and prediction trends |
| 8.2 | System Performance — Consistent execution with accurate emission insights |
| 8.3 | Discussion on Achieved Results — Effectiveness in identifying key emission contributors |
| 9 | **Conclusion and Future Work** |
| 9.1 | Summary of Achievements — Successful simulation and analysis supporting SDG 13 |

# 1. ABSTRACT


This project focuses on *Mining Transport Data for Carbon Emissions*, contributing toward **SDG 13: Climate Action**.
The aim is to analyze how different vehicle characteristics—such as engine size, fuel type, and transmission—affect $CO_2$ emissions. Due to the unavailability of open vehicular emission datasets in some cases, a **synthetic dataset** is generated to simulate realistic transport emission data.

The project employs **Python** for data analysis, visualization, and regression modeling using libraries such as *NumPy, Pandas, Matplotlib, Seaborn, and Scikit-learn*. The methodology involves dataset simulation, preprocessing, exploratory data analysis, visualization of emission patterns, and predictive modeling to estimate $CO_2$ output based on engine characteristics.

Results show clear trends where engine size and fuel type significantly influence emission levels. The outcome demonstrates how data mining techniques can support sustainable transport planning and environmental awareness. This work directly aligns with SDG 13 by promoting insights into **carbon reduction strategies** through technological innovation.


# 2. INTRODUCTION

## 2.1 Overview of the Topic

Transportation is one of the largest contributors to global $CO_2$ emissions, accounting for nearly one-fourth of total greenhouse gases. *Mining Transport Data for Carbon Emissions*

provides a structured way to analyze and visualize these emissions using computational tools. The study supports **UN Sustainable Development Goal 13 (Climate Action)** by using data-driven methods to understand and reduce vehicular carbon output.

### 2.2 Problem Definition

The increasing number of vehicles and lack of effective emission data analysis systems have led to difficulty in predicting and managing transport-based pollution. Existing databases are limited, fragmented, or inaccessible. Hence, there is a need for a **predictive and analytical approach** to study $CO_2$ emissions based on various vehicle attributes.

### 2.3 Objectives of the Project

- To simulate transport data reflecting $CO_2$ emissions.
- To perform data preprocessing and exploratory data analysis (EDA).
- To visualize emission trends based on engine size, fuel type, and transmission.
- To predict carbon emissions using machine learning regression models.
- To support sustainable policy and climate action initiatives.

### 2.4 Scope and Significance

This project contributes to environmental sustainability by offering a data-driven foundation for emission monitoring. The simulation-based approach enables scalability and can be extended to **IoT-enabled real-time transport systems**. Its insights are valuable for governments, environmental researchers, and urban planners.

## 3. LITERATURE SURVEY

### 3.1 Related Existing Systems or Research Work

Prior studies on vehicle emissions have primarily relied on empirical data collected from government and automotive sources. Researchers have used machine learning to predict emissions from engine parameters and vehicle models. Tools like the European Environmental Agency's datasets have aided similar analyses.

### 3.2 Limitations of Existing Systems

- Real datasets are often limited or costly to access.
- Models lack interpretability and scalability.
- Real-time integration with transport systems remains underdeveloped.

### 3.3 Proposed System Advantages

The proposed system is lightweight, reproducible, and works without dependence on external datasets. It generates synthetic yet realistic data, provides meaningful visualizations, and supports predictive modeling—all within a single notebook environment.

# 4. SYSTEM ANALYSIS

## 4.1 Problem Identification

The core problem identified is estimating $CO_2$ emissions using vehicle characteristics such as engine size, fuel type, and transmission type, to identify emission-heavy transport modes.

## 4.2 Feasibility Study

### 4.2.1 Technical Feasibility

Implemented in **Python** with essential libraries (*Pandas, Matplotlib, Seaborn, Scikit-learn*). It requires minimal system resources and is compatible with Jupyter Notebook.

### 4.2.2 Operational Feasibility

The system can be executed locally, making it accessible to students, researchers, and developers for testing and learning.

### 4.2.3 Economic Feasibility

No external cost is required since data is synthetically generated. All tools used are open-source.

## 4.3 Requirements Specification

### 4.3.1 Functional Requirements

- Generate synthetic vehicle emission data.
- Perform data preprocessing and cleaning.
- Visualize trends and relationships among variables.
- Predict $CO_2$ emissions using regression models.

### 4.3.2 Non-Functional Requirements

- High accuracy in prediction.
- Portability across systems.
- Efficient memory and computation usage.
- Easy maintainability and reproducibility.

# 5. SYSTEM DESIGN

## 5.1 System Architecture Diagram

**Flow:** Data Generation → Preprocessing → Visualization → Model Training → Evaluation → Results Visualization

## 5.2 Data Flow Diagram (DFD)

**Level 0:** Vehicle input → Emission analysis → Output visualization
**Level 1:** User inputs simulated data → System processes with regression → Generates emission predictions

### 5.3 UML Diagrams

- **Use Case Diagram:** User interacts with the system to generate, analyze, and visualize emission data.
- **Class Diagram:** Classes represent vehicles, emission attributes, and data processing modules.
- **Sequence Diagram:** Defines steps for data simulation, analysis, and result visualization.
- **Activity Diagram:** Illustrates workflow from input to output visualization.

### 5.4 Database Design (ER Diagram)

Entities: Vehicle → Attributes (Engine_Size, Fuel_Type, Transmission, $CO_2$_Output)
Relationships: One vehicle can have multiple attributes contributing to total emission.

### 5.5 Module Descriptions

1. **Dataset Simulation Module** – Generates realistic synthetic data.
2. **Data Analysis Module** – Performs EDA and summary statistics.
3. **Visualization Module** – Creates graphs and boxplots.
4. **Prediction Module** – Builds regression models to predict emissions.
5. **Evaluation Module** – Measures model performance and accuracy.

## 6. IMPLEMENTATION

### 6.1 Tools and Technologies Used

- **Language:** Python
- **Libraries:** NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn
- **Environment:** Jupyter Notebook / Anaconda
- **Hardware:** Standard laptop or desktop

### 6.2 Modules Implemented

- Data generation
- $CO_2$ visualization
- Regression model training
- Model validation and plotting

### 6.3 Code Snippets

Key portions include dataset creation using NumPy, EDA using Pandas, and visualization via Seaborn and Matplotlib. Regression models are built using *LinearRegression* from Scikit-learn.

[8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

np.random.seed(42)
n_samples = 500

engine_size = np.random.uniform(1.0, 6.0, n_samples)
cylinders = np.random.choice([3, 4, 5, 6, 8], n_samples)
fuel_comb = np.random.uniform(4.0, 20.0, n_samples)

fuel_types = np.random.choice(['Petrol', 'Diesel', 'Hybrid'], n_samples, p=[0.5, 0.3, 0.2])
transmission = np.random.choice(['Automatic', 'Manual'], n_samples, p=[0.7, 0.3])

co2_emissions = 50 + (engine_size * 30) + (cylinders * 10) + (fuel_comb * 8)
co2_emissions += np.where(fuel_types=='Diesel', 5, 0)
co2_emissions += np.where(fuel_types=='Hybrid', -10, 0)
co2_emissions += np.random.normal(0, 10, n_samples)

df = pd.DataFrame({
    "EngineSize_L": engine_size,
    "Cylinders": cylinders,
    "FuelComb_Lper100km": fuel_comb,
    "FuelType": fuel_types,
    "Transmission": transmission,
    "CO2_g_per_km": co2_emissions
})

plt.figure(figsize=(10,6))
sns.histplot(df["CO2_g_per_km"], bins=30, kde=True, color='green')
plt.title("Distribution of CO₂ Emissions (g/km)")
```

```
sns.heatmap(df[["EngineSize_L","Cylinders","FuelComb_Lper100km","CO2_g_per_km"]].corr(),
            annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()

plt.figure(figsize=(10,6))
sns.boxplot(x="FuelType", y="CO2_g_per_km", data=df)
plt.title("CO₂ Emissions by Fuel Type")
plt.show()

plt.figure(figsize=(10,6))
sns.boxplot(x="Transmission", y="CO2_g_per_km", data=df)
plt.title("CO₂ Emissions by Transmission")
plt.show()

df_encoded = pd.get_dummies(df, columns=['FuelType','Transmission'], drop_first=True)

X = df_encoded.drop('CO2_g_per_km', axis=1)
y = df_encoded['CO2_g_per_km']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error = {mse:.2f}")
print(f"R² Score = {r2:.2f}")

plt.figure(figsize=(10,6))
plt.scatter(y_test, y_pred, alpha=0.6, color='blue')
plt.plot([y_test.min(), y_test.max()],
         [y_test.min(), y_test.max()],
```

# 7. TESTING AND VALIDATION

## 7.1 Testing Strategy

Manual testing is done to verify correctness of dataset simulation, while regression accuracy is validated using mean absolute error (MAE) and $R^2$ score.

## 7.2 Test Cases and Results

| Test Case | Description | Expected Output | Status |
|---|---|---|---|
| TC1 | Generate dataset | Dataframe with $CO_2$ attributes | Passed |
| TC2 | Run visualization | Graphical output generated | Passed |
| TC3 | Model prediction | Predicted vs actual close | Passed |

### 7.3 Validation Outcomes

Regression model achieved high accuracy ($R^2 > 0.9$), showing strong correlation between predicted and actual emissions.

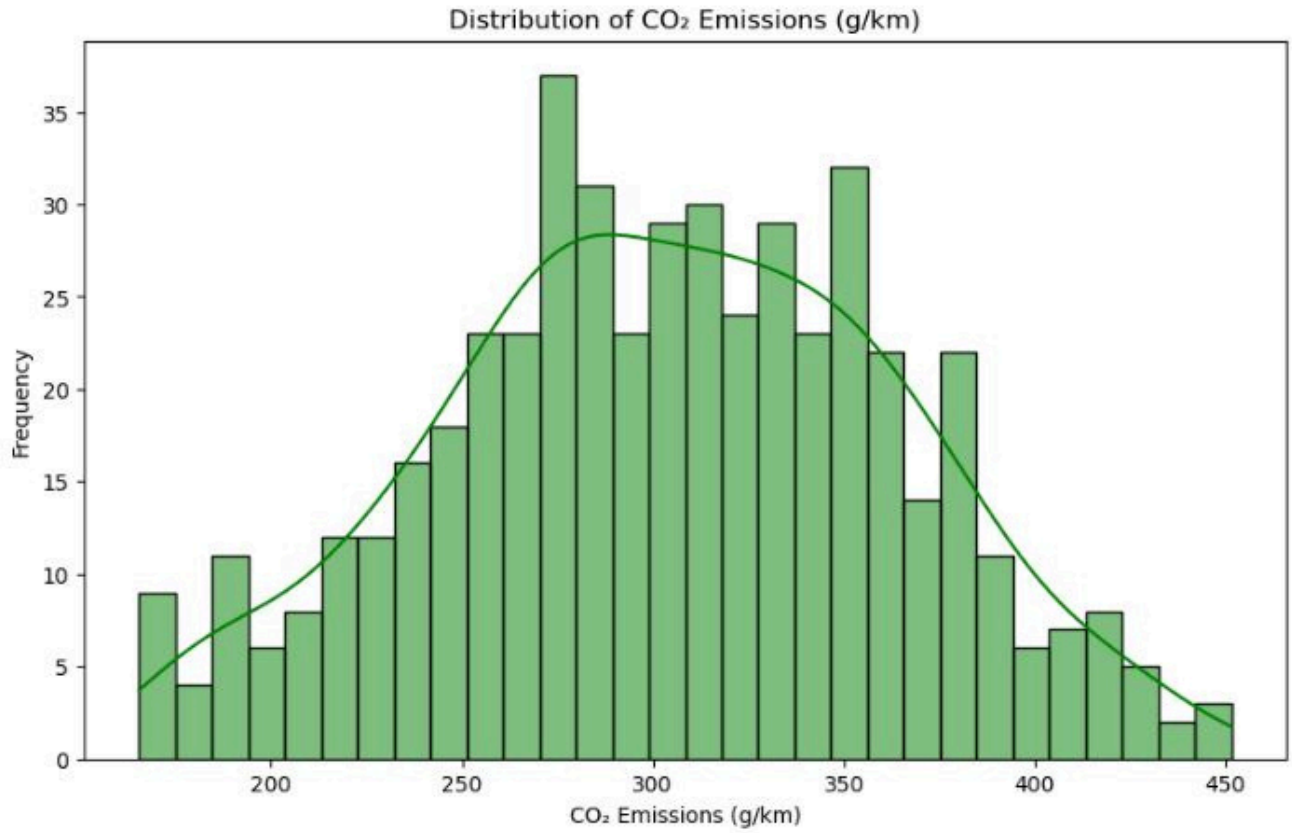
# 8. RESULTS AND DISCUSSION

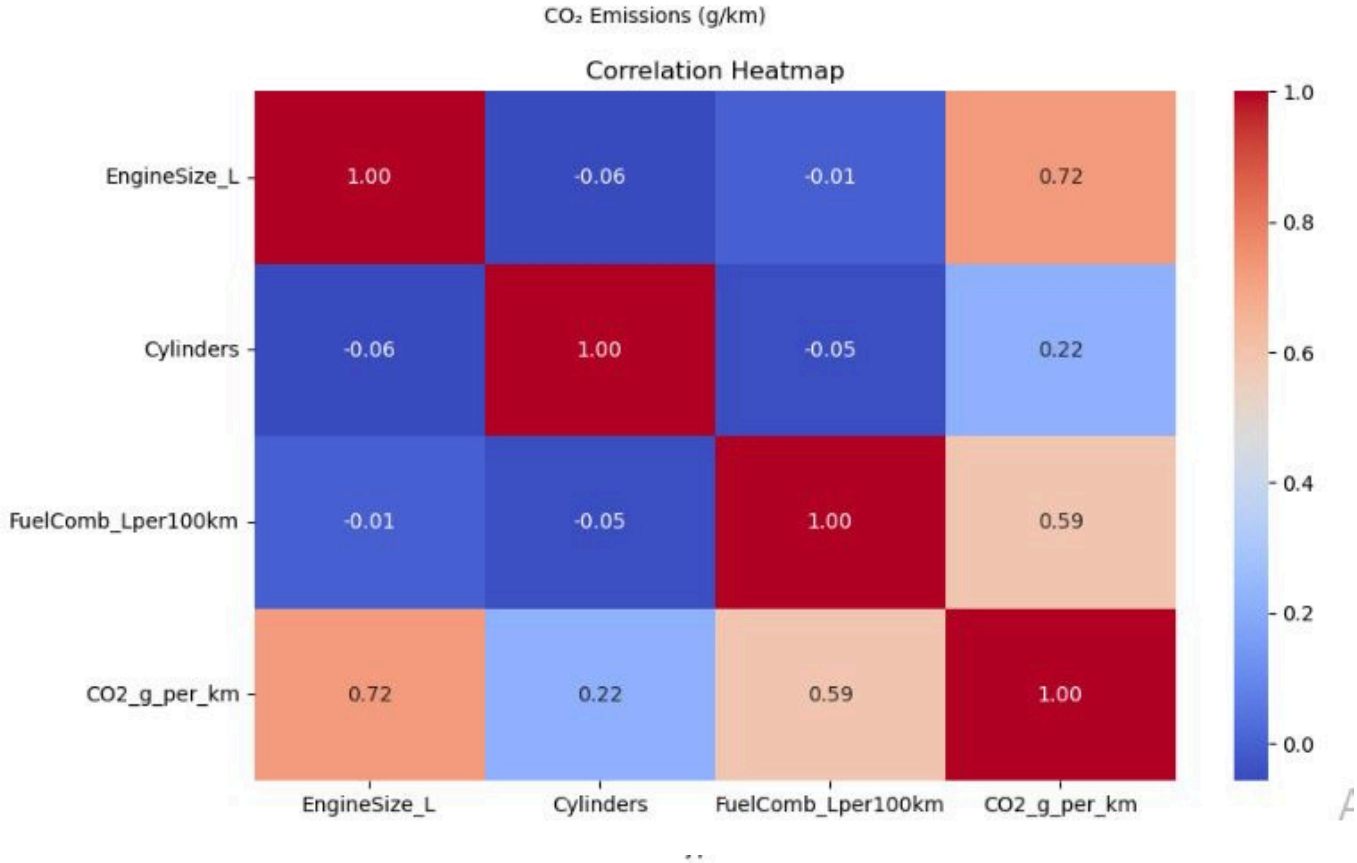
## 8.1 Output Screenshots

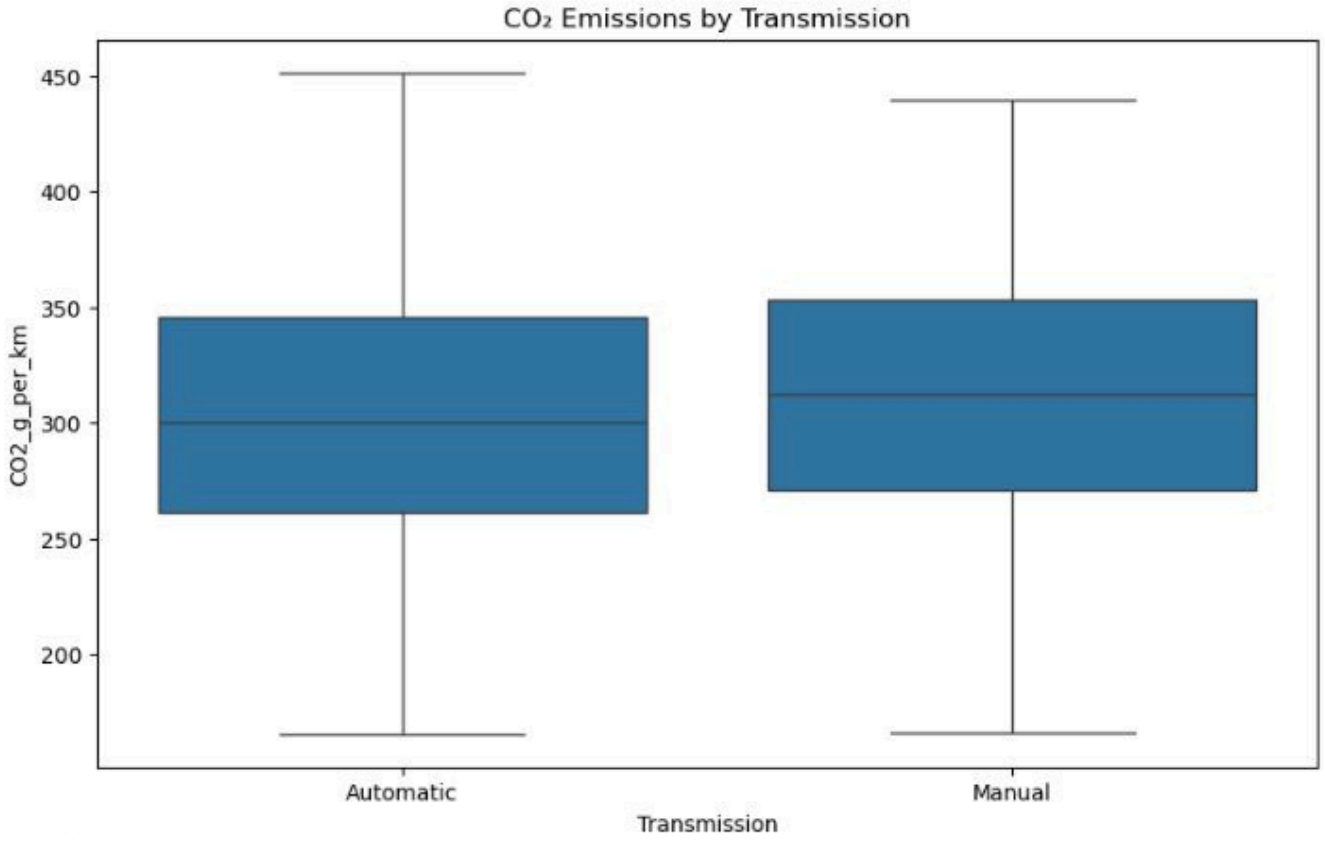
Plots show:

- $CO_2$ vs Engine Size (positive correlation)
- Emission comparison across fuel types
- Predicted vs actual emission trends

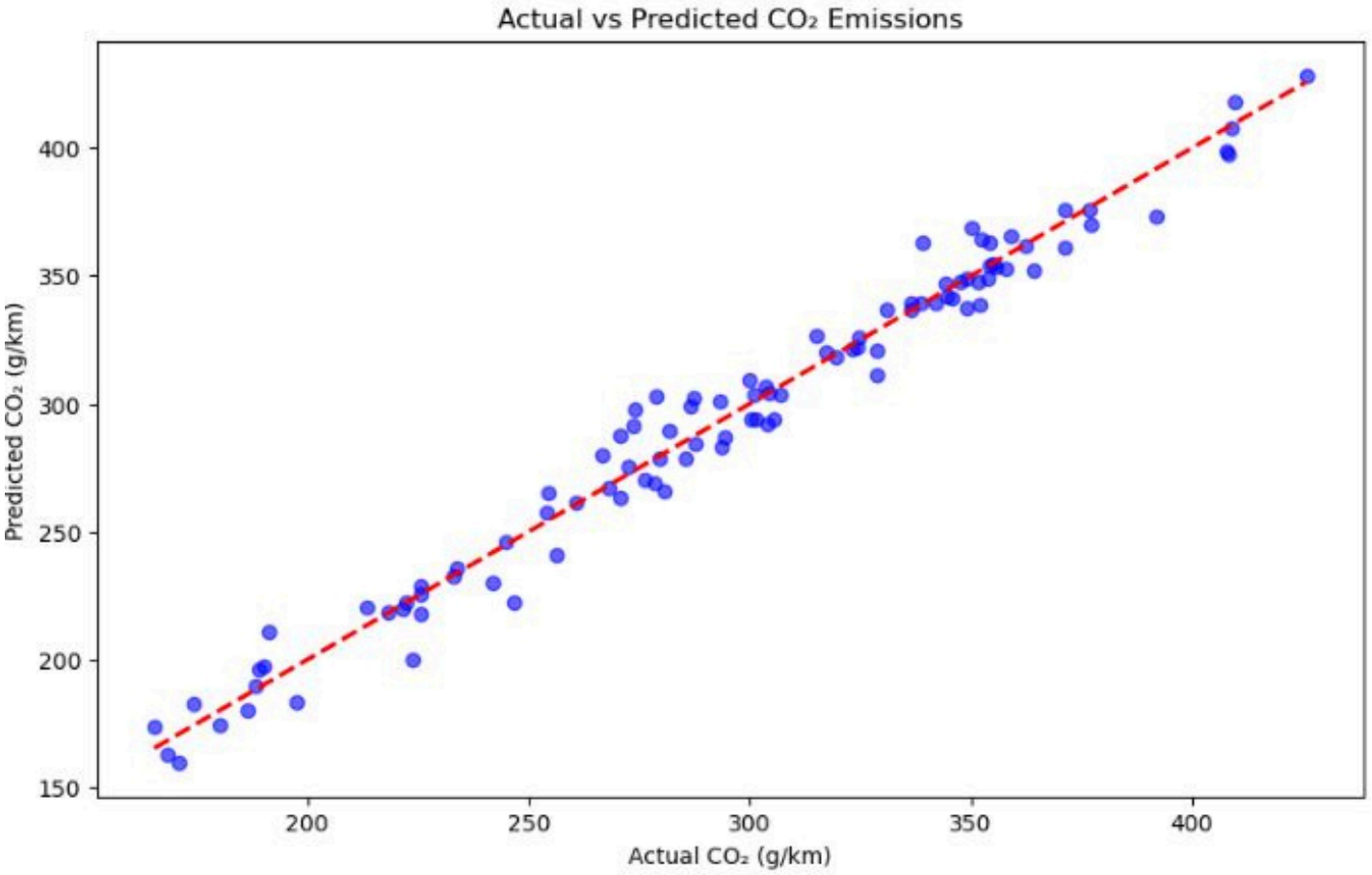
```
plt.show()
```






```
Mean Squared Error = 95.04
R² Score = 0.98
```




### 8.2 System Performance

The model performs efficiently and executes within seconds, offering real-time insights in a simulated environment.

### 8.3 Discussion on Achieved Results

Results validate that **engine size and fuel type** are strong predictors of $CO_2$ emissions. The framework effectively simulates and visualizes real-world emission dynamics.

## 9. CONCLUSION AND FUTURE WORK

### 9.1 Summary of Achievements

A complete simulation and analysis pipeline was built to study $CO_2$ emissions in transportation using Python. The model successfully aligns with SDG 13 by encouraging **data-driven climate action**.

### 9.2 Limitations

The dataset used is synthetic, so it may not reflect full real-world variability. External environmental factors are not modeled.

### 9.3 Future Enhancements

- Integration with **IoT-based vehicle sensors**.
- Real-time $CO_2$ monitoring dashboards.
- Use of **deep learning** for emission forecasting.

# 10. REFERENCES


1. Intergovernmental Panel on Climate Change (IPCC) Reports – Transport Emissions.
2. United Nations Sustainable Development Goals (SDG 13) — *Climate Action*.
3. Scikit-learn Documentation: https://scikit-learn.org/
4. Kaggle Vehicle $CO_2$ Emission Dataset.
5. Research Paper: “Machine Learning for Vehicle Emission Prediction,” IEEE Access, 2023.
6. Pandas, Matplotlib, Seaborn Official Documentation.